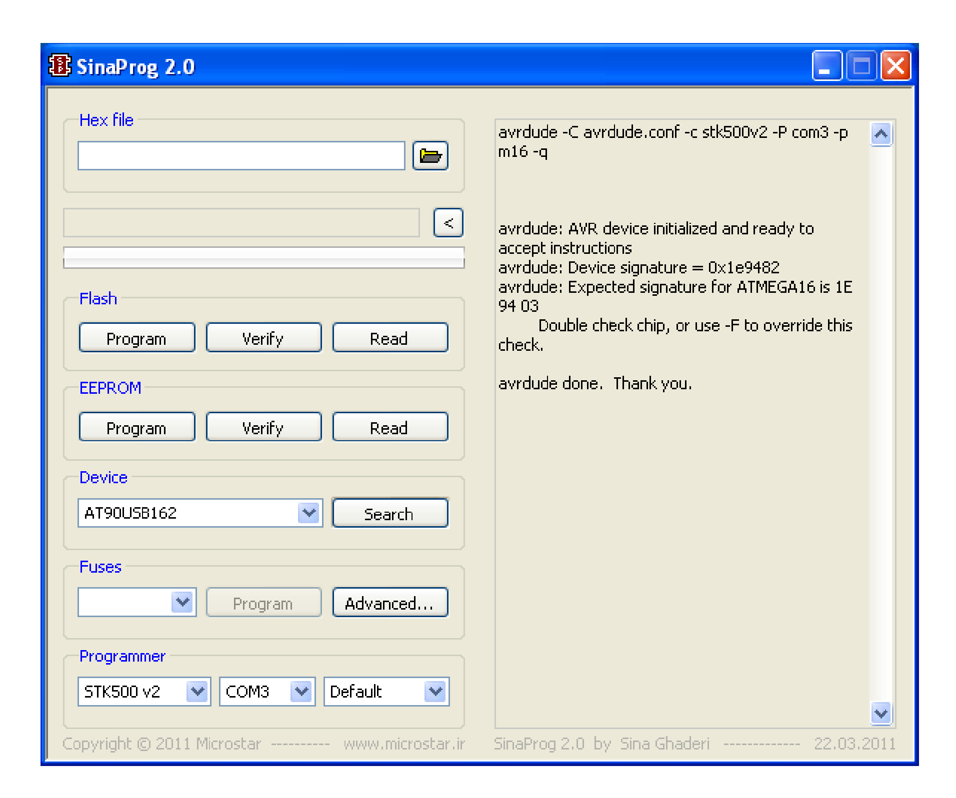




Programando con SinaProg

Manual de usuario para programar microcontroladores Atmel AVR en Windows con *SinaProg*



Naxus Sistemas Inteligentes
Cra. 84A # 5 - 12
Cali, Colombia

www.naxuscolombia.com
contacto@naxuscolombia.com

Revisión 1.0
Agosto 2011



Índice de contenido

1. INTRODUCCIÓN

SinaProg es un programa Freeware para Windows que implementa una interfaz gráfica para la programación de microcontroladores AVR de Atmel utilizando el reconocido software AVRDUDE. Dada la descontinuación del software *WinAVR* desde Enero de 2010, el cual presentaba como un IDE completo y de código abierto para usuarios de Windows que incluía AVRDUDE, SinaProg aparece como una excelente alternativa actualizada para la programación de microcontroladores AVR en Windows. SinaProg permite realizar operaciones de lectura, programación, verificación y borrado de las memorias *FLASH* y *EEPROM* de los dispositivos AVR, así como la programación de sus *Fusebits* y *Lockbits*, todo a través de una interfaz sencilla y simple pero muy práctica.

2. INSTALACIÓN DE SINAPROG

El primer paso para utilizar SinaProg es instalarlo en su computador. No existe una página oficial de descarga. Recomendamos descargar la versión que se encuentra en la página de *Naxus Sistemas Inteligentes* en caso de que utilice el programador NXProg, (http://www.naxuscolombia.com/descargas-comunes/). Sin embargo se dejan a continuación algunos enlaces habilitados para la versión 2.0 para usuarios de otros programadores hardware.

- http://www.filecrop.com/sinaprog.html
- http://rapidshare.com/files/454283222/SinaProg_2.0.zip

El archivo que se obtiene en la descarga no es un instalador, es una carpeta comprimida. Se recomienda descomprimirla directamente en la carpeta *Archivos de Programa*, donde generalmente se encuentran los programas utilizados en Windows, como se puede ver en la imagen 1.

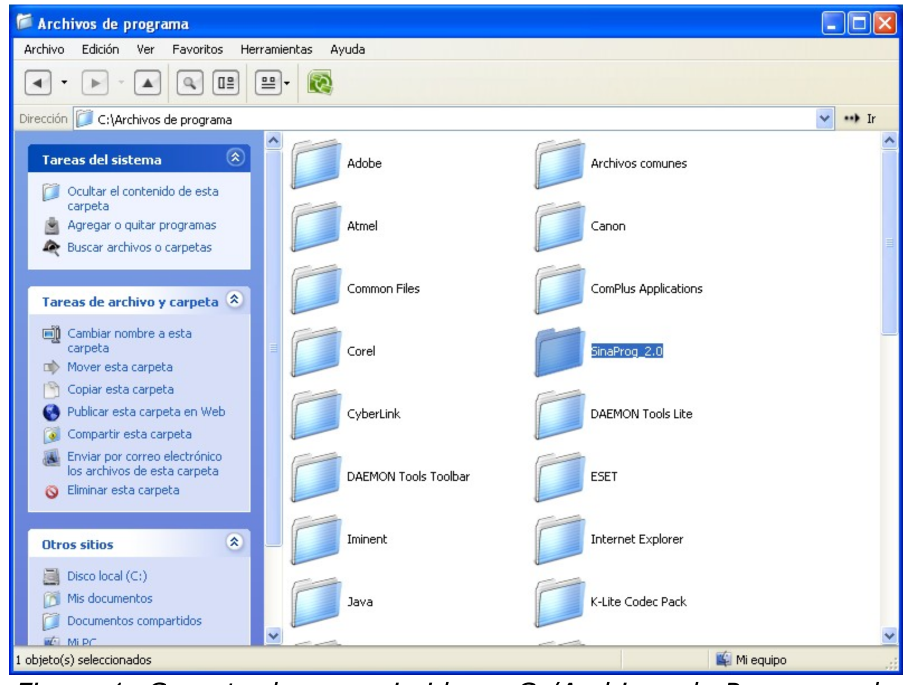


Figura 1: Carpeta descomprimida en C:/Archivos de Programa de Windows





Solo con descomprimir la carpeta ya puede ejecutar Sinaprog con el archivo .EXE que contiene esta carpeta.

2.1. Creando una entrada en el menú Inicio de Windows

Para acceder a SinaProg desde el *menú Inicio* de Windows realizamos el siguiente procedimiento: Creamos un hipervínculo al archivo ejecutable (llamado *sinaprog.exe*) utilizando clic derecho y seleccionando la respectiva opción en el menú que se despliega, como se muestra en la imagen 2.

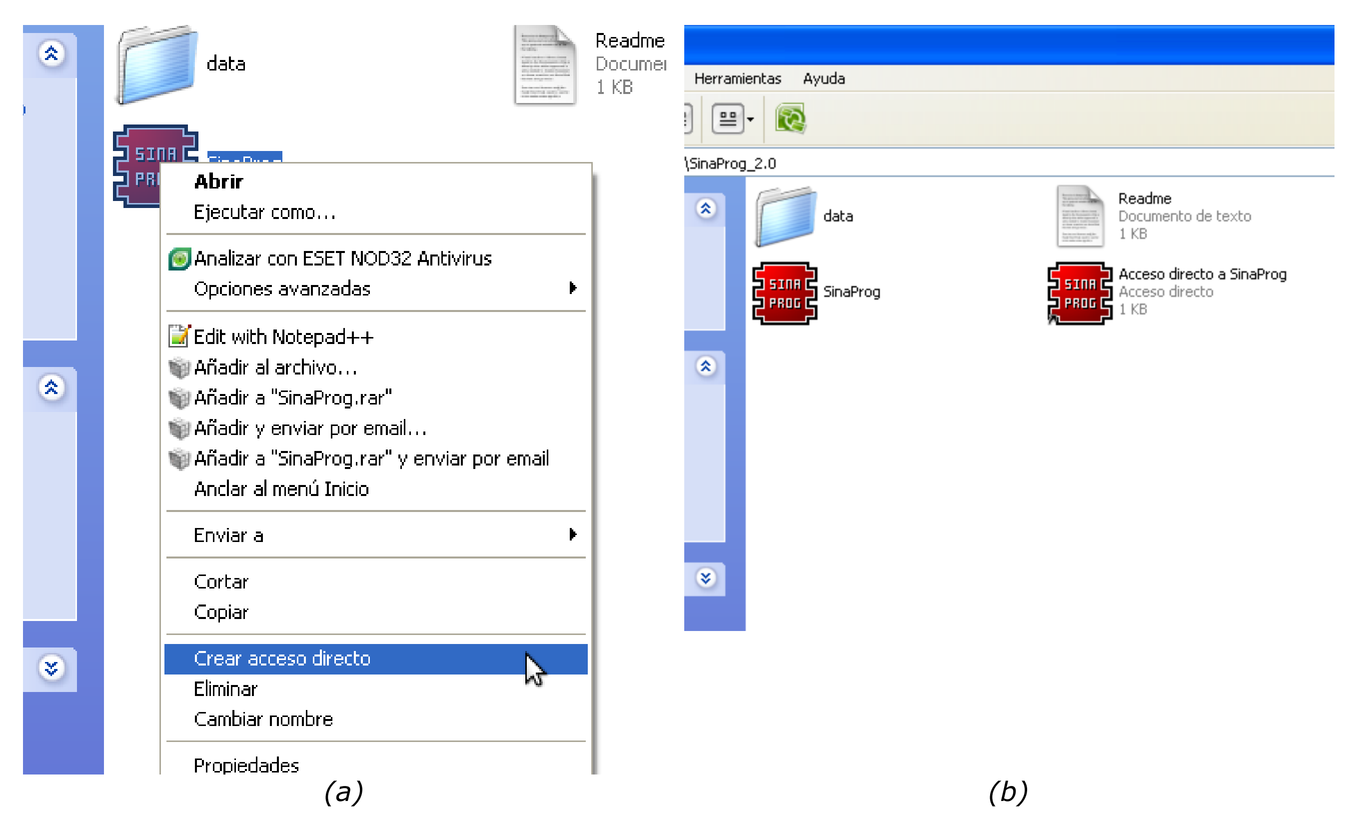


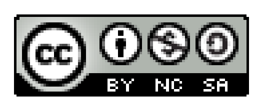
Figura 2: Crear acceso directo a la aplicación

Se observa que se crea un hipervinculo denominado "*Acceso directo a Sinaprog*". El hipervínculo recién creado lo cortaremos y pegaremos en nuestra carpeta de usuario en de *Documents and Settings.* Creamos una carpeta llamada *Sinaprog* que genera un submenu en el *menú Inicio* de Windows, y ahí colocamos el hipervínculo, renombrándolo a *Sinaprog*. Esto se puede observar en la figura 3.

Figura 3: Creando el acceso directo en el menú Inicio

De esta manera queda completamente instalado Sinaprog en nuestro sistema y podemos ejecutar el programa desde el *menú Inicio* de Windows.





3. UTILIZANDO SINAPROG

Cuando ejecutamos Sinaprog llamándolo desde el *menú Inicio* de Windows (figura 4), aparece la interfaz mostrada en la figura 5, en donde se detallan cada una de sus secciones.



Figura 4: Ejecución del Sinaprog desde el menú Inicio de Windows

3.1. Interfaz del programa

La interfaz de Sinaprog queda descrita en la figura 5.



Figura 5: Interfaz del programa Sinaprog



Como se observa, contamos con las siguientes secciones:

- **Selección de archivo HEX**: En esta sección elegiremos el archivo hexadecimal generado en la etapa de compilación, que será programado en la memoria del microcontrolador.
- **Mensajes**: Utilizando este botón podemos observar los mensajes que arroja Sinaprog por cada tarea realizada. La figura 6 ilustra como se vería la interfaz con esta opción habilitada.
- **Programación de memoria FLASH**: Podemos programar, verificar y leer la memoria Flash del dispositivo.
- **Programación de memoria EEPROM**: Podemos programar, verificar y leer la memoria EEPROM del dispositivo.
- **Selección del dispositivo**: En esta sección podemos elegir el dispositivo que deseamos programar manualmente, o podemos realizar una búsqueda para que el programador encuentre qué dispositivo se encuentra conectado.
- **Configuración de Fuses**: En esta sección podemos reprogramar los Fusebits y Lockbits del microcontrolador.
- **Selección de programador**: Podemos elegir el programador hardware del que dispongamos. Se encuentran incluídos la mayoría que soporta AVRDUDE.

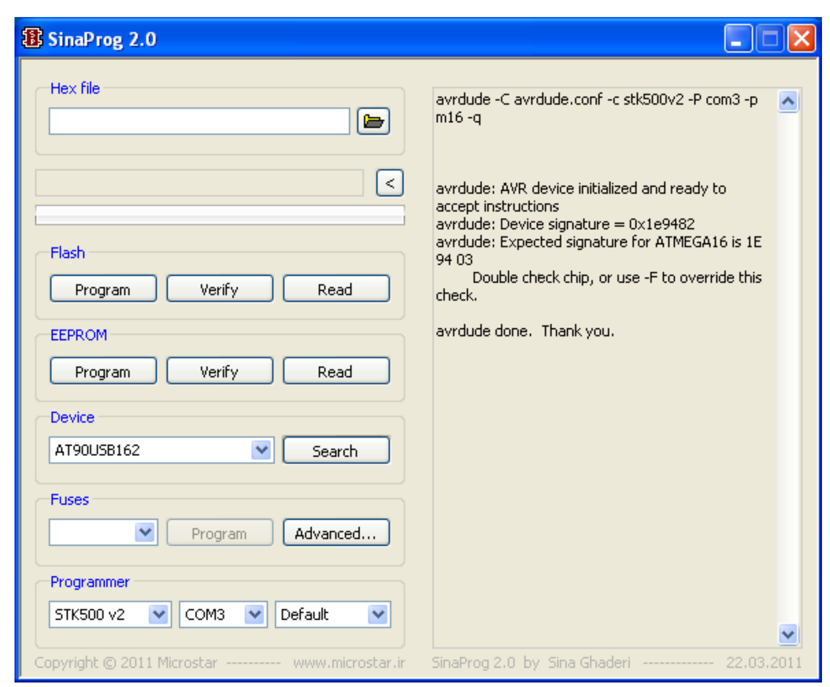


Figura 6: Interfaz de Sinaprog con la sección de mensajes habilitada.





3.2. Selección del programador

El primer paso para realizar la programación es seleccionar el programador hardware que utilizaremos. De la lista desplegable en la sección de programador, seleccione el idóneo. Verifique primero que su programador sea efectivamente compatible con AVRDUDE (ver figura 7.a)



NOTA: Si cuenta con NXProg como programador Hardware, seleccione la opción *STK500v2* de la lista desplegable.

A continuación seleccione el puerto donde se encuentre conectado el programador, en caso de que éste sea reconocido como un puerto serial. En caso negativo, deje este menú con la opción *Default* habilitada (ver figura 7.b). Por último, seleccione la velocidad en baudios de la transmisión, en caso de tratarse de un programador serial. La opción *Default* es la recomendada.



NOTA: Para NXProg se recomienda utilizar la opción *Default* o *BR 115200*.

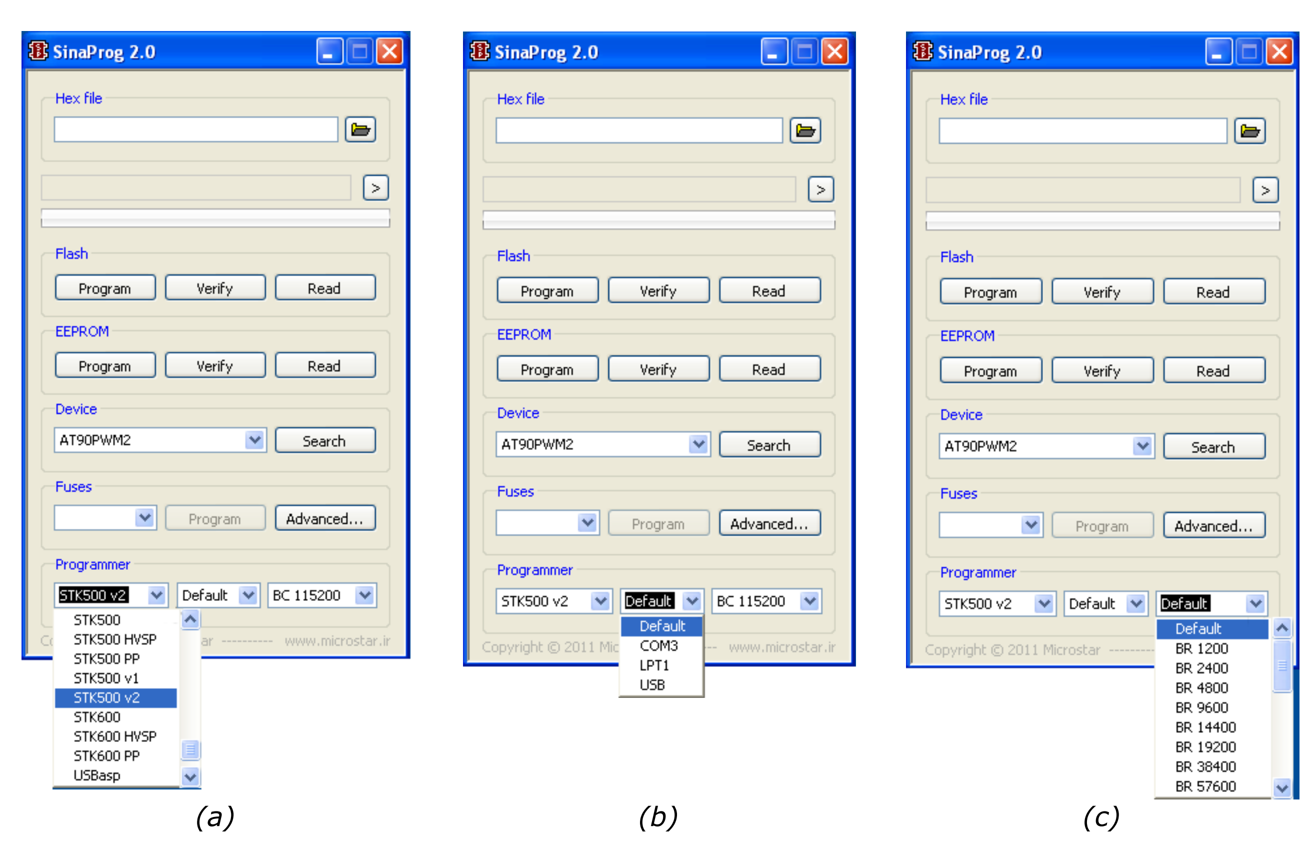
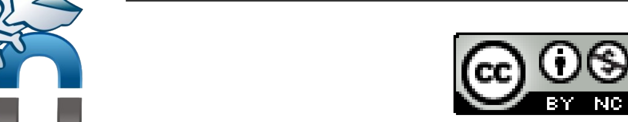


Figura 7: Configuración del programador utilizado





3.3. Selección del dispositivo

Con el programador seleccionado procedemos a seleccionar el dispositivo que nos interesa programar. Para ello podemos bien sea escogerlo de la lista desplegable de la sección respectiva, o podemos utilizar el botón *Search* (figura 8) para que el programador identifique qué dispositivo se encuentra conectado. Por obvias razones es necesario tener conectado correctamente el programador al microcontrolador para usar esta opción.

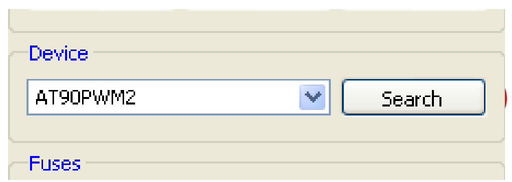


Figura 8: Botón Search de la sección de dispositivo

3.4. Configuración de Fuses

La configuración de *Fusebits* y Lockbits del microcontrolador se realiza pulsando el botón *Advanced...* de la sección de *Fuses*, como se observa en la figura 9

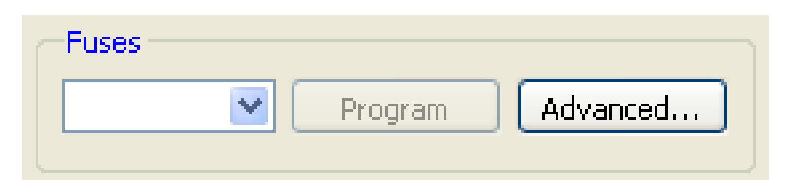
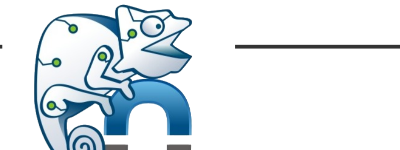
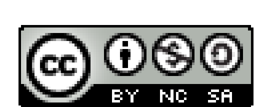


Figura 9: Sección de Fuses, botón Advanced

Se abrirá la ventana mostrada en la figura 10. En esta podemos modificar los *Fusebits*, lo que corresponde al byte alto, el bajo y el extendido (en caso de que el microcontrolador cuente con este byte), así como los *Lockbits*.





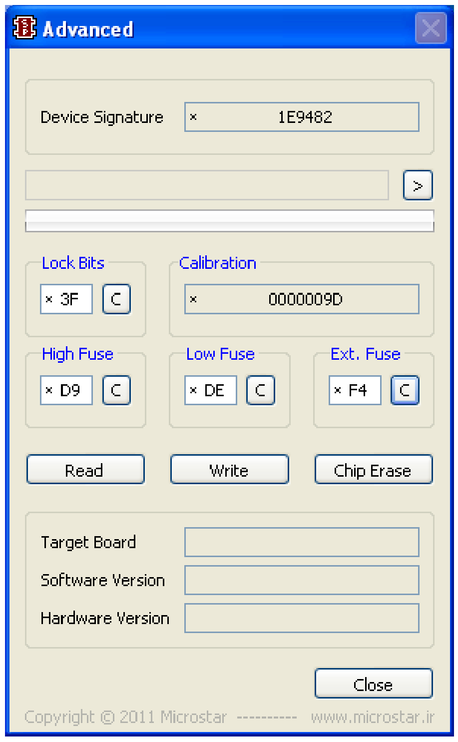


Figura 10: Interfaz de programación de Fuses

Para configurar cada byte, se utiliza el botón [C] que aparece en cada opción. A continuación se describen las ventanas que aparecen para cada opción.

3.4.1. Programación de Lockbits

Los *Lockbits* son una protección que proveen los microcontroladores Atmel AVR para proteger la memoria del dispositivo contra lecturas de memoria no deseadas. Es posible definir el nivel de protección de la memoria Flash de la *sección de Aplicación* así como la *sección de Bootloader*. Refierase a la hoja de datos de su dispositivo para mayor información. La programación de *Lockbits* se realiza mediante la ventana de la figura 11. En el primer menú desplegable se seleccionan las opciones generales (si se desea o no protección mediante *Lockbits*). El segundo menú desplegable define el nivel de protección de la *sección Aplicación* de la memoria Flash. Finalmente, el tercer menú desplegable define el nivel de protección de la *sección Bootloader* de la memoria Flash.

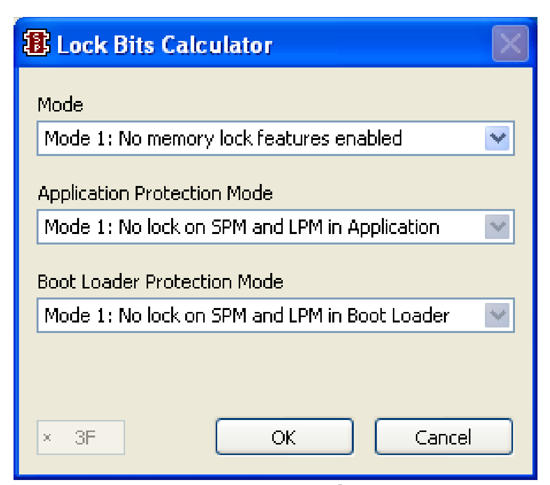


Figura 11: Programación de Lockbits

3.4.2. Programación de Fusebits

Los *Fusebits* definen el comportamiento de algunos periféricos de los microcontroladores Atmel AVR, como la fuente de reloj (interna, cristal, RC), interfaces de programación por defecto (JTAG, SPI), preservar datos de memoria EEPROM, entre otros. Refiérase a la hoja de datos de su dispositivo para información más específica. La programación de los *Fusebits* se realiza definiendo los valores de cada byte en las respectivas ventanas mostradas en la figura 12. No se describirán aquí en detalle ya que son específicas para cada microcontrolador.

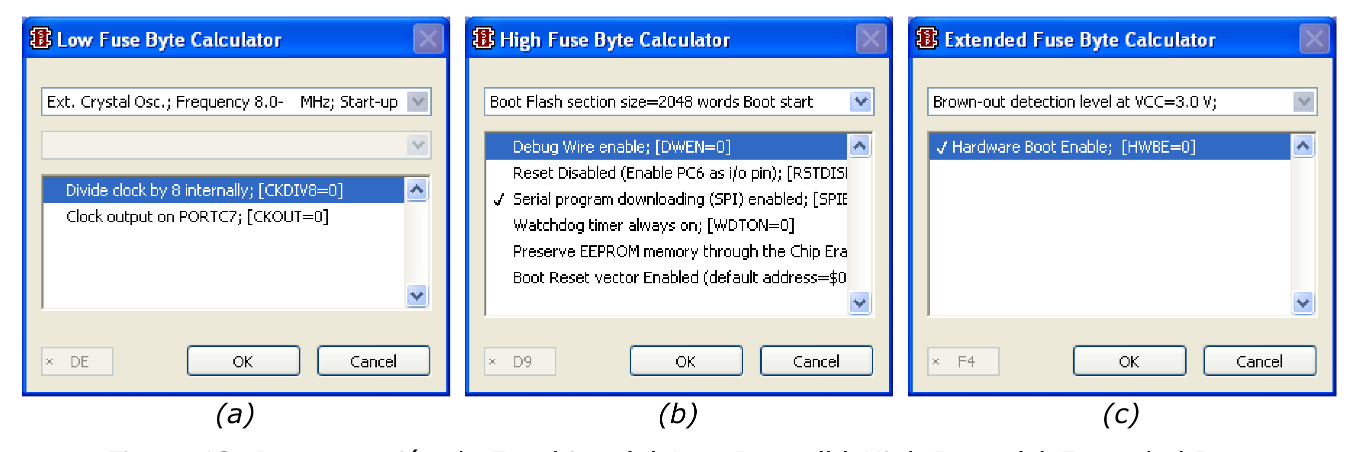
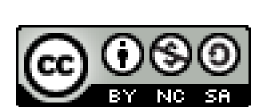
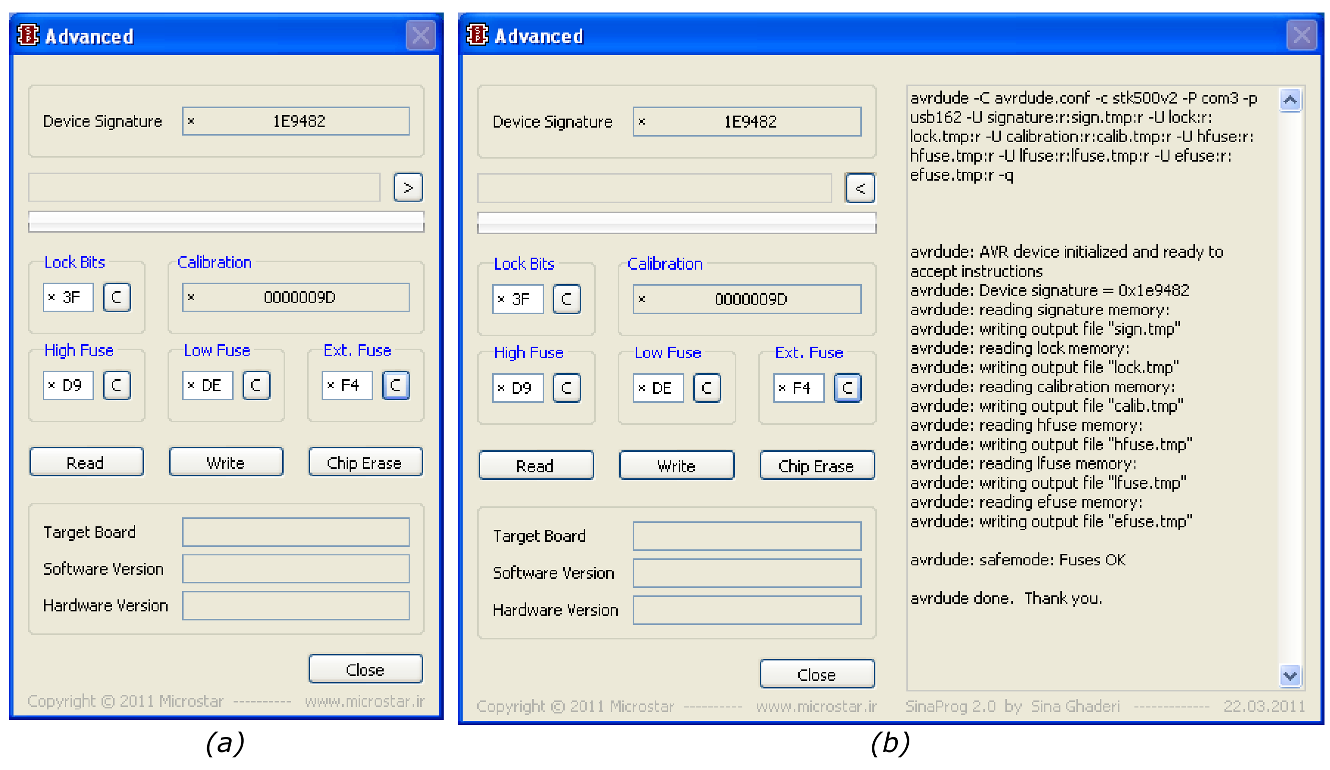


Figura 12: Programación de Fusebits: (a) Low Byte, (b) High Byte, (c) Extended Byte





Cuando termine de realizar la configuración deseada de los *Fuses*, proceda a programarlos usando el Botón Write. Recuerde que puede leer la configuración actual de Fuses del dispositivo usando el botón Read. Asimismo, recuerde que puede usar el botón > para observar los mensajes que arroja el programador en los ciclos de lectura/escritura de los *Fuses*, como se muestra en la figura 13.



(a) *(b)*

Figura 13: Ventana de programación de Fuses (a) sin mensajes y (b) extendida con mensajes del programador

3.5. Selección del archivo HEX

El archivo HEX representa los datos en formato binario que serán programados en la memoria Flash del dispositivo. Es el archivo generado por el compilador utilizado en la etapa de desarrollo del software, sin importar si ha escrito su aplicación en ASM o en C/C++. En la sección de la interfaz relacionada con esta tarea, encuentra el botón que le permite buscar este archivo en su computador. Seleccione el deseado y de clic en *Aceptar*. La figura 14 ilustra este procedimiento.

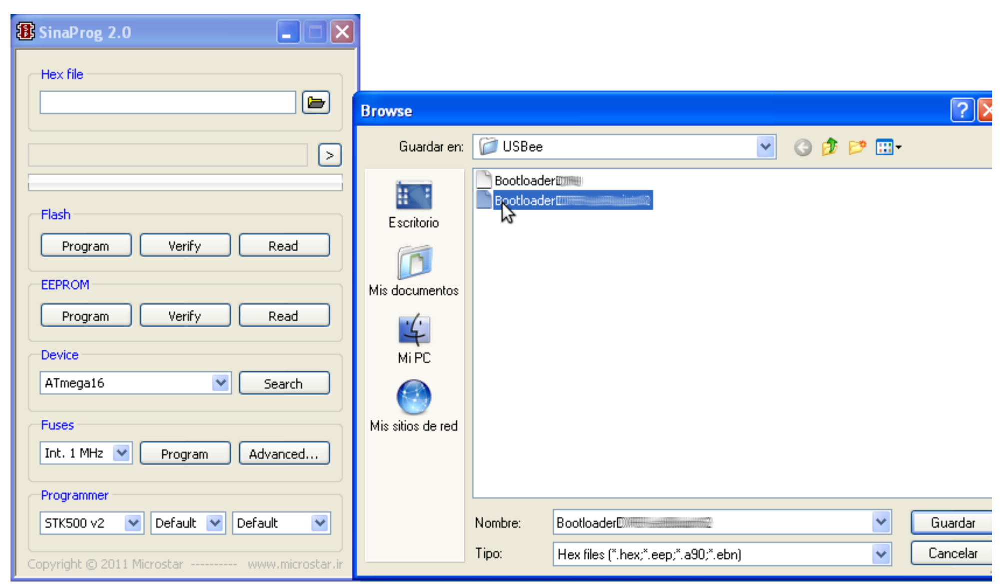


Figura 14: Seleccionando el archivo HEX

3.6. Programación de FLASH/EEPROM

La programación de memorias FLASH o EEPROM del microcontrolador se realizan en la respectiva sección de la interfaz, como se observa en la figura 15. Use los botones *Program* para programar el microcontrolador. Esta operación realiza primero un ciclo de borrado previo a la programación. Use el botón *Verify* para validar que el estado de la memoria Flash corresponde con el del archivo HEX anteriormente cargado. El botón *Read* le permite leer la memoria Flash del dispositivo y almacenarla en un archivo HEX. Recuerde sin embargo que esta opción será posible siempre y cuando al microcontrolador que vaya a leer no tenga los *Lockbits* programados. En caso de ser así, sólo es posible leer la memoria luego de haber realizado una operación de borrado de la misma.

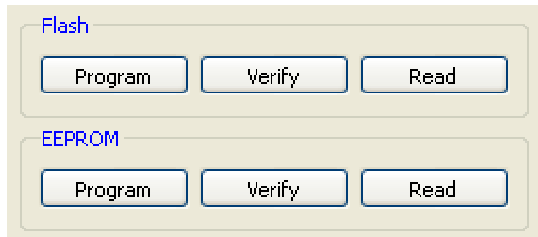


Figura 15: Secciones de programación de memoria FLASH y EEPROM

